CS230

Real-Time Seismic Attributes Computation with Conditional GANs

João Paulo Navarro
Stanford
Departament of Computer Science
jppno@stanford.edu

Abstract

In this work, we propose a deep learning pipeline for training neural networks that can accurately approximate attributes in three-dimensional seismic datasets. Using generative adversarial modeling, we train specialized networks that learn to map attributes, given an amplitude seismic volume as input. The trained network is used to compute transformations in original data much faster than its exact formulation since inference time is rapid in modern GPU architectures. Initial results show that deep neural networks are robust to learn different attributes with distinct data distributions. Via model inference, attribute computations is up to 80x faster than classical formulation.

1 Introduction

One major goal in seismic data analysis is the detection of geologically relevant structures. Subsurface properties, such as faults, channels and other events contain valuable spatio-temporal information with scientific and commercial importance. There is a compelling need in the identification of these features in an accurate and efficient way so that the chronostratigraphy of the area of interest is well understood. The underground analysis can be significantly improved by the use of automatic or semi-automatic interpretation tools. In this context, seismic attributes contribute to the process, highlighting regions that may be hard to visually identify, reducing decision-making time.

A seismic attribute is any measure of seismic data that helps us visually enhance or quantify features of interpretation interest. A good attribute is either directly sensitive to the desired geologic feature or reservoir property of interest or allows us to define the structural or depositional environment [1]. Mathematically, a seismic attribute is a mapping $A_n(D) \in \mathbb{R}^3$, given an input amplitude volume $D \in \mathbb{R}^3$ (Figure 1). The formulation of transformation A_n depends upon the characteristics that aimed to be enhanced in the data. For instance, if we want to identify sharp edges, one may apply the Sobel filter [2], with a similar mathematical definition used in image processing.

A challenge in attribute calculation is processing time. Seismic surveys may reach terabytes, and when particular transformations involves sophisticated formulations (e.g. frequency domain attributes need Fourier transforms), computing time may significantly increase, taking hours to process the entire dataset.

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

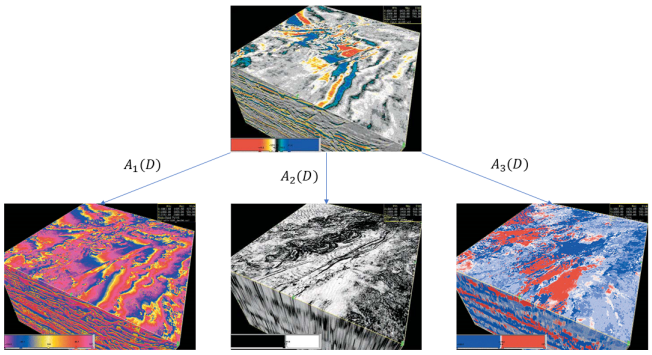

Figure 1: Examples of seismic attributes: original amplitude volume (D) serves as input for A_1, A_2, and A_3 transformations.

We are proposing a deep learning (DL) training pipeline based on conditional Generative Adversarial Networks (conditional GANs) [3] to learn the mapping pattern of attributes. One network is trained for each A_n mapping. These models are used for calculations (inference) at unseeing volumes. The main advantage of this approach is the fact that we have a linear time to compute any type of seismic attribute, depending only on volume size.

2 Related work

Machine learning (ML) models are being used by the geosciences community for a while. The most common algorithms are classical unsupervised approaches like K-Means, Principal Component Analysis (PCA), and Self-organizing Maps (SOM) to categorize traces waveforms [4]. We also find in literature ML seismic attributes used to enhance faults, channels and other discontinuities with unsupervised learning methods [5]. More recently, DL approaches are being applied to a range of different applications in geophysics, such as fault detection with convolutional neural networks (CNN) [6], full-waveform inversion [7], and relative geological time attributes [8]. Researchers are also exploring more sophisticated models, such as GANs, to improve seismic data resolution [9] and perform noise attenuation [10].

3 Dataset and Features

Seismic datasets are usually computationally represented as 3D scalar fields, being each voxel of the volume a 32-bits floating point (amplitude). This data has visual coherence, and we can understand it as RMI scans of Earth's subsurface. The Society of Exploration Geophysics (SEG, [11]) and the U.S. Geological Survey (USGS, [12]) made available online repositories with major surveys from onshore and offshore areas around the globe. We collected seismic datasets stacked by different processing methods from different regions, as depicted in Table 1.

Dataset Name	Geography	Size (GB)	Grid Dimension
Poseidon3D	Australia	5.2	$(583 \times 2351 \times 945)$
Parihaka3D	New Zealand	3.9	$(870 \times 1040 \times 1080)$
SantaYnez3D_1	United States	2.1	$(389 \times 1074 \times 1260)$
SantaYnez3D_2	United States	1.5	$(262 \times 1163 \times 1176)$
NorneFiled2006	Netherlands	1.1	$(321 \times 1001 \times 851)$

Table 1: Used seismic surveys.

SantaYnez3D_2 and NorneField2006 were selected as development datasets. Others were reserved for training. We processed all surveys using the open-source interpretation software OpenDetect (OD) [13]. This processing generates transformed volumes, which serve as outputs for our supervised learning pipeline. We have chosen attributes with distinct data distributions. For each survey in

Table 1, we generated the respective Energy, Instantaneous Phase and Coherence attributes [1]. After cropping the volumes to select most relevant areas, and computing the attributes, training and development set summed up a grand total of 55.2 GB, stored as SEG-Y [14] files.

4 Method

In this work, we are interested in developing a generative framework that could be used to replace direct seismic attributes calculations. To reach this goal, some requisites must be met: a) real-time or interactive-time inference is desired, b) pixel-wise approximations with small errors (10^{-3} or less), c) one trained network per attribute, and d) network must perform attribute computations respecting the ground-truth as much as possible, avoiding creating new patterns, even with same data distribution.

To achieve the requirements, we use as baseline the conditional GANs proposed by Wang et. al. [15]. This network operates on high-definition image patches, which helps saturate the PCIe bus between CPU-GPU, increasing the inference performance, which is important to achieve real-time inference.

4.1 Architecture

The network consists of conditional GANs, with coarse-to-fine generators $G = \{G_1, G_2\}$ and multi-scale discriminators $D = \{D_1, D_2, D_3\}$. In our task, the objective of the generator G is to translate input seismic images to attributes, while the combined discriminator D aims to distinguish real attributes from translated ones. The framework operates in a supervised setting: the training dataset is given as a set of pairs of corresponding images $\{(\mathbf{s_i}, \mathbf{a_i})\}$, where $\mathbf{s_i}$ is the input, and $\mathbf{a_i}$ is the respective attribute. Conditional GANs, applied to our problem, aim to model the conditional distribution of attributes, given an input seismic image via the following minimax game [15]:

$$\min_G \max_{D_1,D_2,D_3} \sum_{k=1,2,3} \mathcal{L}_{GAN}(G, D_k), \tag{1}$$

where the objective function $\mathcal{L}_{GAN}(G, D_k)$ is given by:

$$\mathbb{E}_{(\mathbf{s},\mathbf{a})}[\log D_k(\mathbf{s}, \mathbf{a})] + \mathbb{E}_{\mathbf{s}}[\log(1 - D_k(\mathbf{s}, G(\mathbf{s}))]. \tag{2}$$

Coarse-to-fine-generator The generator is decomposed into two sub-networks, G_1 and G_2. The term G_1 refers to the global generator network and G_2 corresponds to the local enhancer. Specifically, the input to the residual blocks in G_2 is the element-wise sum of feature map from G_2 and the last feature map from G_1. This generator design promotes an effective aggregation between global and local information during the image synthesis task.

Multi-scale Discriminators There are some important challenges training GAN discriminators in high-resolution image synthesis. To differentiate real and synthesized images, the discriminator needs a large receptive field. This leads to deeper networks with larger convolution kernels, which may generate model overfitting. To deal with this problem, 3 discriminators with identical network structures are used. Each discriminator operates in different lower-resolution scales, improving the ability of final discriminator $D = \{D_1, D_2, D_3\}$ to distinguish real and synthesized samples.

4.2 Implementation Details

The original network was built to translate semantic labels into photo-realistic images. We adapt the original network as follows: a) we changed initial layers to receive 1-channel 32-bits images, instead of discrete semantic labels, b) removed the restriction to feed the network with boundary maps and instance maps, and c) disabled the instance-level control, used to diversify the image synthesis for a given semantic label map.

Since we are working with training samples having data distribution different from ordinary natural images, all networks are trained from scratch. Thus, performance is a crucial factor. One major bottleneck in training pipelines at multi-GPU servers is data ingestion and manipulation. We have implemented a high-performance parallel data ingestion engine that extracts random samples from SEG-Y files. The volume reader was built on top of Equinor's Segyio library [16]. Random crops

with fixed size (512×512) are extracted in parallel from the original data and passed to the GPU. In GPU, we perform the following data augmentations: polarity inversion (seismic specifics), flip, rotation, scaling, and normalization.

5 Experiments and Results

In this section, we present the obtained results training different conditional GANs used to translate seismic images to attributes. For each type of attribute, we have trained dozens of networks using servers with 8x NVIDIA Tesla V100 GPUs and select the best models. Each training cycle takes up to 2 days to run. All implementations were done using the DL framework PyTorch [17].

To ensure maximum GPU usage and thus a lower training time, we have set the batch-size as the higher possible value, which is 160. Therefore, the GPU memory occupancy reaches a peak of 99%. Seismic domain images have lower data variability, in comparison with the universe of natural images. The fact we are dealing with 1-channel images reinforce this hypothesis. Therefore, optimizing a seismic-to-attribute network is simpler than a label-to-image task. We verified that none of the used metrics improved after 100 epochs. Thus, we reduce the total number of epochs to 100. We kept the same learning rate for the first 50 epochs and linearly decay the rate to zero over the next 50 epochs.

As a comparison baseline, we have reproduced one of the latest works on DL to seismic attributes and compared them with our approach. Geng et. al. [8] proposed a U-Net architecture using residual blocks and refinement convolutions to map the Relative Geological Time (RGT) attribute. Since no source code was released, we have implemented a version of their work in Keras [18], strictly following the instructions provided in the paper. Code is under directory `dsa_final/rgt`. In Table 2, we show metrics comparing the proposed GAN architecture (Ours) with the aforementioned work (RGT). Results show consistent superiority of GAN over a deep U-Net for regression.

In Table 2 (Ours), we also show metrics collected from the development set: mean squared error (MSE), peak signal-to-noise ratio (PSNR), and structural similarity (SSIM). The MSE and PSNR correlate numerical differences between the original and synthesized attributes, while the SSIM emphasizes the perceptual divergences. From the numerical point of view, the worst result was achieved by the Instantaneous Phase network. This dataset presents a high-variance in voxel intensity (high-frequencies), which is naturally harder to capture, even for more advanced DL models. Furthermore, the high variability in values of neighboring voxels contributed (in absolute value), to increase the MSE, which the PSNR depends on.

Attribute	MSE (RGT)	PSNR (RGT)	MSE (Ours)	PSNR (Ours)	SSIM (Ours)
Semblance	1×10^{-2}	17.06	1×10^{-3}	29.64	0.94
Phase	6×10^{-2}	11.65	9×10^{-3}	20.31	0.93
Energy	1×10^{-3}	28.8	8×10^{-5}	40.82	0.98

Table 2: Quantitative results on development set (SantaYnez3D_2 & NorneField2006). We compare our results with the RGT architecture proposed by Geng et. al. [8].

When we visually compare the real and synthetic samples in Figure 2, we verified that differences are hard to identify. All networks were able to capture the high-level features satisfactorily. This leads to the conclusion that global generator G_2 performed well for all cases, but local enhancer had difficulty to properly approximate narrow high-frequencies (Fig. 2, right-side). One way to improve the low-level feature generation is appending a third sub-network G_3. Thus, we would starting training the new coarse generator G_3 at images with (128×128) resolution, and gradually increase the feature map size to (256×256) and (512×512), training G_2 and G_1.

We use a single Tesla V100 with 32GB to perform inference experiments. In Table 3 we depict the total time to compute the attributes using its exact formulation with OpenDetect (OD) [13], and inference with the proposed network. Attribute calculus with OD ran on a machine with 12 cores. The time to compute attributes with OD takes, on average, 56.22 seconds for NorneField2006 and 68.11s for SantaYnez3D_2 dataset, while it only takes 0.70s and 0.86s, respectively, with our DL approach. The speedup achieved was around 80x times. This real-time performance may be a gaming-changing, when attributes over terabytes of data must be computed.

Dataset	Phase	Energy	Semblance	OD Average Time	Ours	Speedup
NorneField2006	59s	25s	84s	56.22s	0.70s	**80.47**
SantaYnez3D_2	67s	27s	110s	68.11s	0.86s	**79.57**

Table 3: Attribute computation using OpenDetect and the proposed work. Network inference is 80x faster than conventional approaches.

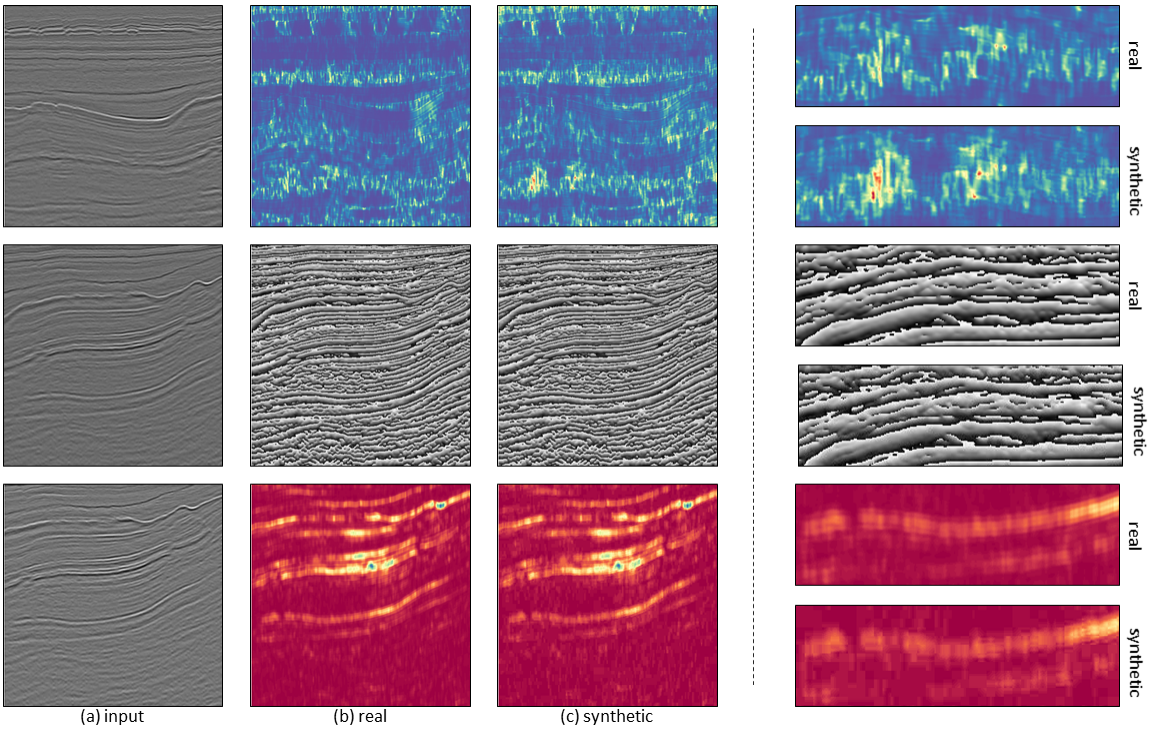

Figure 2: Results of conditional GANs trained to compute Semblance (top), Instantaneous Phase (middle) and Energy (bottom) seismic attributes. The overall image structure is well captured. Local features present minor differences in expanded regions of interest (right panels).

We get to the conclusion that conditional GANs can successfully approximate the computation of seismic attributes, with high visual fidelity, and minor numerical differences. The presented architecture may operate as a universal attribute calculator, with some restrictions. For a high-quality numerical reconstruction, we must specialize the model and properly adjust the architecture and tune the hyperparameter space to fit each data. This effort may be worthwhile if we have an attribute that its exact computation takes hours, and processing time needs to decrease to seconds.

6 Conclusion/Future Work

In this work, we explored the use of conditional GANs to compute seismic attributes in volumetric datasets. The trained networks were used to infer attributes quantities at unseeing surveys, reaching real-time performance. Quantitative and qualitative results show that a single GAN architecture can be satisfactorily trained to learn the mapping of different attributes, but with minor errors. If we want minimal numerical differences between original and reconstructed volumes, we should tune and specialize the model. As future work, we intend to explore the use of 3D convolutions and experiment different network architectures.

References

[1] S. Chopra and K.J. Marfurt. *Seismic Attributes for Prospect Identification and Reservoir Characterization*. Geophysical development series. Society of Exploration Geophysicists, 2007.

[2] Satinder Chopra, Rajive Kumar, and Kurt Marfurt. Seismic discontinuity attributes and sobel filtering. pages 1624–1628, 08 2014.

[3] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[4] Hiren Maniar, Srikanth Ryali, Mandar S. Kulkarni, and Aria Abubakar. *Machine-learning methods in geoscience*, pages 4638–4642. 2018.

[5] J.P. Peçanha, A.M. Figueiredo, Geisa Faustino, E.A. Perez, Pedro Mario Silva, and Marcelo Gattass. Minimal similarity accumulation attribute using dimensionality reduction with feature extraction. 05 2016.

[6] Xinming Wu, Luming Liang, Yunzhi Shi, and Sergey Fomel. Faultseg3d: Using synthetic data sets to train an end-to-end convolutional neural network for 3d seismic fault segmentation. *Geophysics*, 84:IM35–IM45, 02 2019.

[7] Zhendong Zhang and Tariq Alkhalifah. Regularized elastic full waveform inversion using deep learning. *GEOPHYSICS*, pages 1–47, 06 2019.

[8] Zhicheng Geng, Xinming Wu, Yunzhi Shi, and Sergey Fomel. Relative geologic time estimation using a deep convolutional neural network. pages 2238–2242, 08 2019.

[9] Haijing Zhang, Wei Wang, Xiaokai Wang, Wenchao Chen, Yanhui Zhou, Cheng Wang, and Zhonghua Zhao. An implementation of the seismic resolution enhancing network based on gan. pages 2478–2482, 08 2019.

[10] Xu Si, Yijun Yuan, Fangyuan Ping, Yue Zheng, and Liyi Feng. Ground roll attenuation based on conditional and cycle generative adversarial networks. pages 95–98, 01 2020.

[11] SEG Open Data Repository. https://wiki.seg.org/wiki/open_data.

[12] U.S. Geological Survey. https://www.usgs.gov/.

[13] OpenDetect. dGB Earth Sciences. https://www.dgbes.com/.

[14] SEG-Y Format. https://en.wikipedia.org/wiki/seg-y.

[15] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. *CoRR*, abs/1711.11585, 2017.

[16] Equinor Segyio. https://github.com/equinor/segyio.

[17] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. pages 8024–8035. Curran Associates, Inc., 2019.

[18] François Chollet et al. Keras. `https://keras.io`, 2015.